
# Universidad Nacional Autónoma de México


## Facultad de Ingeniería

## Laboratorio de Diseño Digital

---

# Reporte de práctica 4

## Circuito elevador de Bits al cuadrado

---

*Profesor:*
M.I. Guevara Rodríguez Ma. del Socorro

*Alumno(s):*
Francisco Pablo Rodrigo

Grupo: 6

Calificación total _____

Previo _____

Desarrollo _____

Conclusiones _____

11 de marzo de 2019

# Índice

# 1. Objetivos

## 1.1. General

El alumno diseñará circuitos combinacionales.

## 1.2. Particular

El alumno analizará, diseñará e implementará multifunciones algebraicas utilizando distintas formas de optimización.

# 2. Introducción

Los circuitos combinacionales son, como su nombre lo sugiere, circuitos cuya salida depende solamente de la "combinación" de sus entradas en el momento que se está realizando la medición en la salida. Los circuitos de lógica combinacional son hechos a partir de las compuertas básicas compuerta AND, compuerta OR, compuerta NOT. También pueden ser construidos con compuertas NAND, compuertas NOR, compuerta XOR, que son una combinación de las tres compuertas básicas. La operación de los circuitos combinacionales se entienden escribiendo las ecuaciones booleanas y sus tablas de verdad.

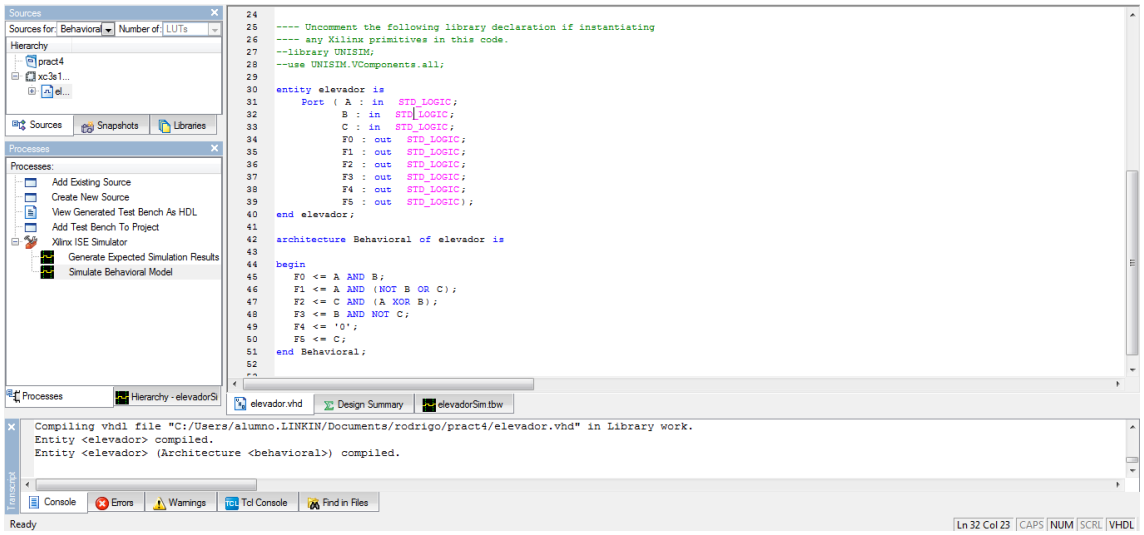
Todos los circuitos combinacionales pueden representarse empleando álgebra de Boole a partir de su función lógica, generando de forma matemática el funcionamiento del sistema combinacional. De este modo, cada señal de entrada es una variable de la ecuación lógica de salida.

Por otra parte, la manipulación de funciones booleana puede llegar a ser muy compleja por lo cual se hace necesario el concepto de minimización. La minimización es básicamente la simplificación de una función, obteniendo una expresión que contenga menos términos o menos variables que la función original. Esto se refleja en la obtención de circuito mas económicos por tener un menor numero de compuertas.

# 3. Previo

# 4. Desarrollo

Para realizar un circuito elevador de 3 bits al cuadrado recurrimos al diseño lógico de Xilinx utilizando la sección de código que a mi parecer es más sencillo que arrastrar los componentes, finalemente el código quedó de la siguiente manera.




Comprobamos que el circuito estuviera bien y para ello realizamos la simulación y probamos para diferentes valores como puede verse a continuación.

$000^2 = 00000$

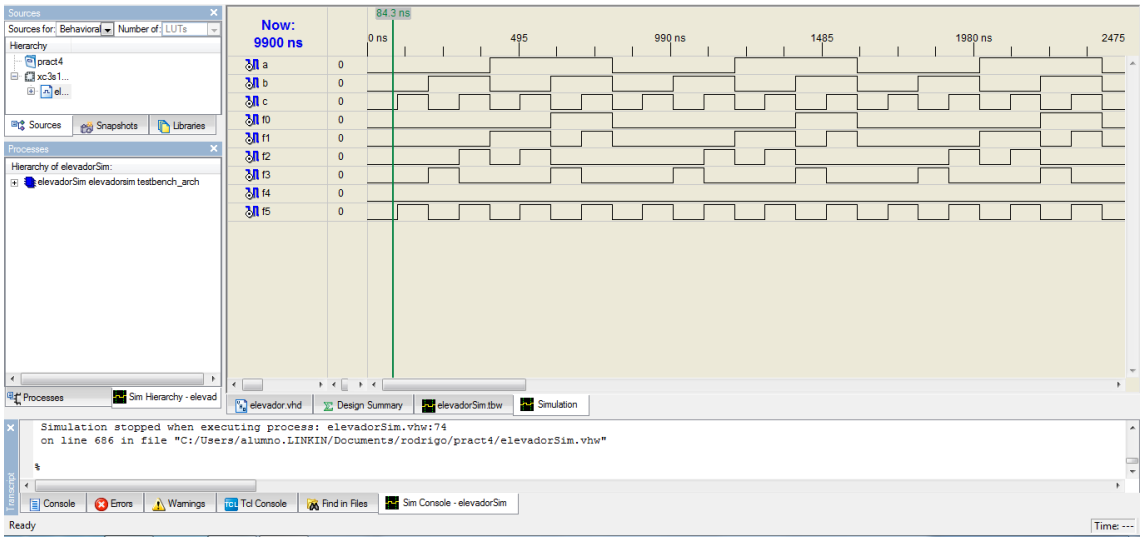



Figura 1: Puede verse que aquí estamos elevando cero al cuadrado y el resultado es cero.

$001^2 = 000001$

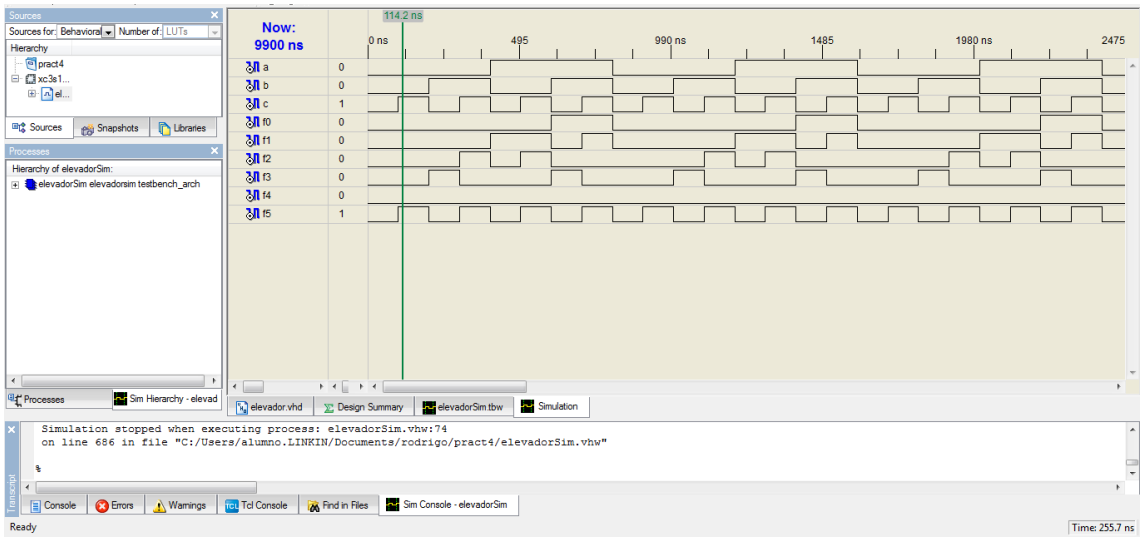



Figura 2: Puede verse que aquí estamos elevando uno al cuadrado y el resultado es uno.

$111^2 = 110001$

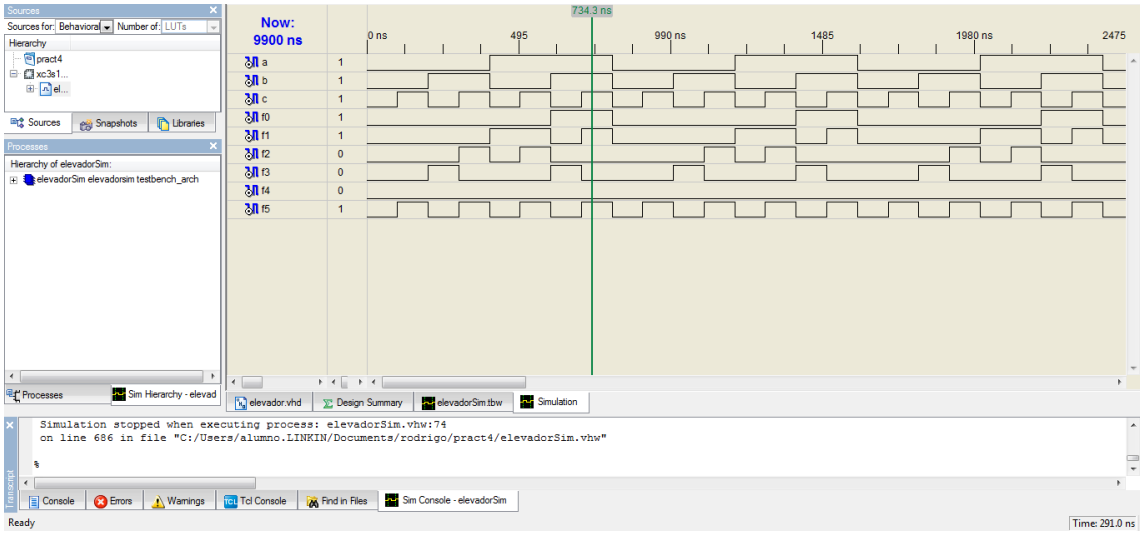



Figura 3: Puede verse que aquí estamos elevando siete al cuadrado y el resultado es 49.

Luego de que la fase de simulación fue exitosa pasamos nuestro diseño a la *Basys2* para ello tuvimos que generar un archivo con extensión *.bit* y posteriormente mediante una programa llamado **Adapt** logramos pasar nuestro circuito a la FPGA.

En la siguiente imagen se puede apreciar a nuestro dieño en funcionamiento. Utilizamos los primeros 6 LEDs y los primeros 3 switches.

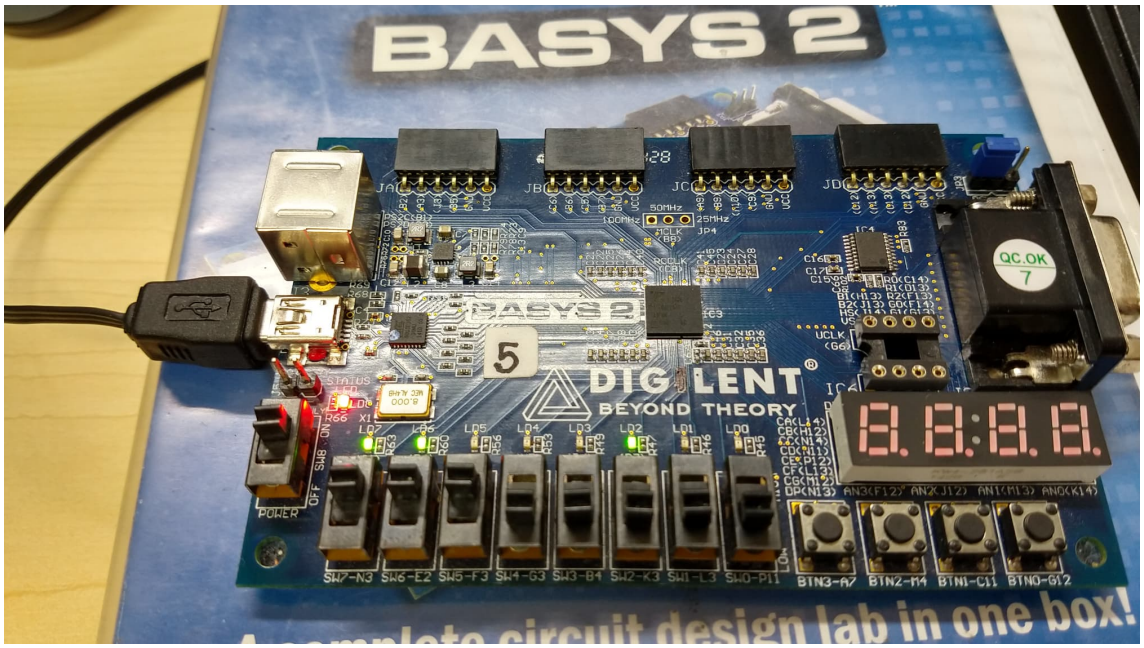



Figura 4: Puede verse que aquí estamos elevando 7 al cuadrado y el resultado es 49.

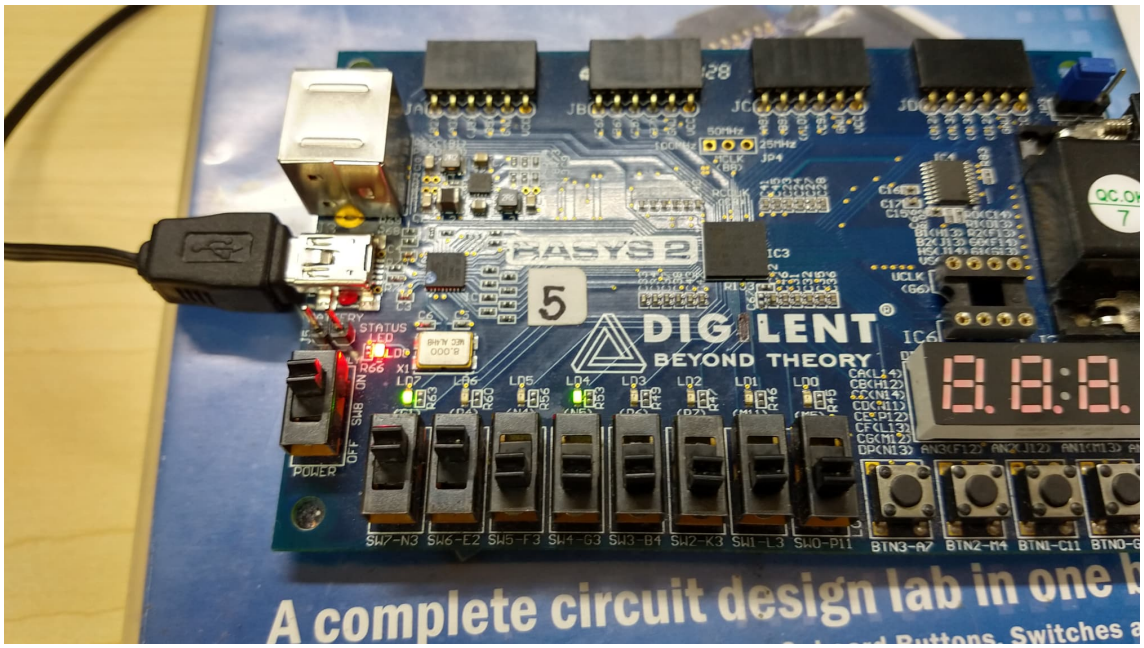



Figura 5: Puede verse que aquí estamos elevando 36 al cuadrado y el resultado es 36.

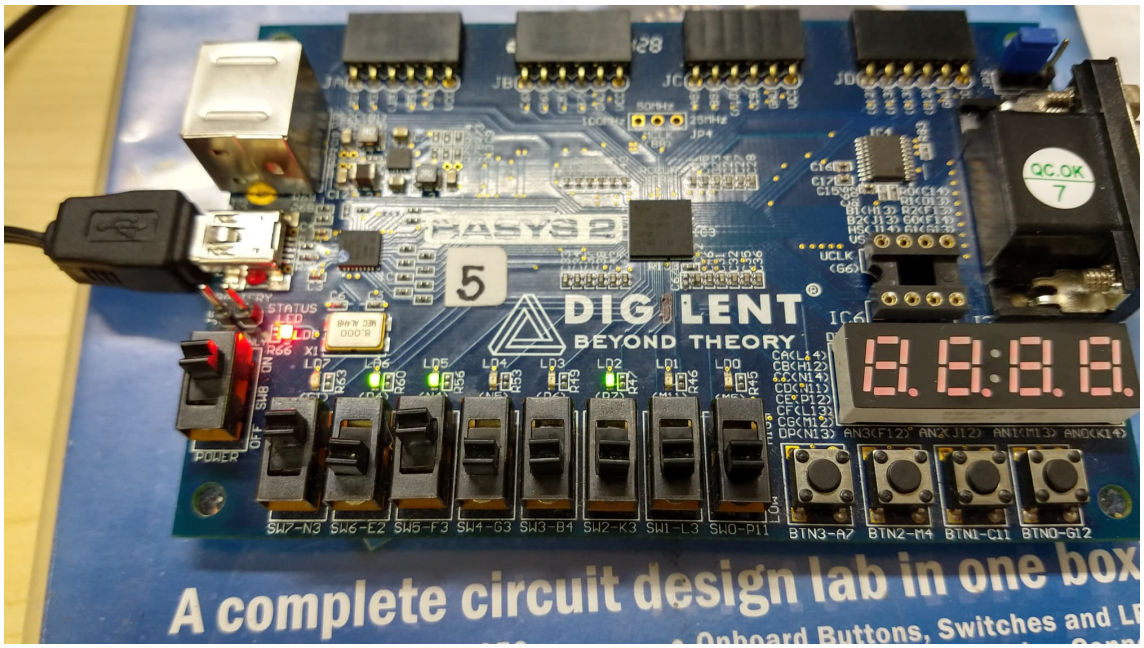



Figura 6: Puede verse que aquí estamos elevando 5 al cuadrado y el resultado es 25.

# 5. Conclusiones

Al realizar el diseño del circuito identifique la importancia de saber minimizar con mapas de Karnaugh ya que sino lo hubiera hecho, el número de compuertas a utilizar sería muchisimo más grande que el que ya tenemos.

Se puede notar que para construir un circuito combinacional que solo puede multiplicar hasta el 7 en decimal o hasta el 111 en binario se llevo 8 compuertas (4 ANDs, 2 NOT, 1 OR y 1 XOR). Quiere decir que para hacer una multitplicación de n por n tendremos que tener muchisimas compuertas, lo cual no es optimo.

Por otra parte, al utilizar el modo de programación del Xilinx observe que es más rápido hacerlo de esa manera en lugar de ir arrastrando componentes y tener unirlos con cuidado.